Phân Tích Dữ Liệu Thực Tế với Python

Bài 03.1: Lập Trình Python Cơ Bản - P2

Quang-Khai Tran, Ph.D
CyberLab, 02/2023

(Ảnh: Internet)

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

CYBERLAB
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Giảng viên và Trợ giảng

Quang-Khai Tran

Postdoctoral Scholar at **KISTI** 한국과학기술정보연구원

Studied Big Data Analytics at **Korea University of Science and Technology**

Facebook: https://www.facebook.com/tqkhai2705/
Email: tqkhai0527@gmail.com

De-Thu Huynh

Giảng viên Thỉnh giảng: TS. Huỳnh Đệ Thủ
Facebook: https://www.facebook.com/dethu.huynh

Trợ giảng

Nguyễn Bùi Hoàng Long

Facebook:
https://www.facebook.com/hoanglong.nguyenbui.96

Nguyễn Trường Thuận

Facebook:
https://www.facebook.com/truongthuannn

Trần Phi Long

Facebook:
https://www.facebook.com/IsaacFA1992

Nội dung

Lập trình Python Cơ Bản - Phần 2:

1. Cơ bản về List/Array trong Python
2. Đọc/ghi dữ liệu
3. Các phân tích thống kê đơn giản
4. Bài tập & Thảo Luận

Phần 1. Cơ bản về List/Array

1.1 Giới thiệu

Một “List” (danh sách) là một biến (hoặc một cấu trúc dữ liệu) được dùng để lưu nhiều giá trị cùng lúc

- Syntax: nằm giữa 2 dấu ngoặc vuông [ ]. Ví dụ:

```
danh_sach1 = [1, 2, 3, 4, 5]
danh_sach2 = [1.1, 2.2, 3.3, 4.4, 5.5]
danh_sach3 = [True, False, True, True, False]
ds_học_viên = ["Nam", "Lan", "Mai", "Việt"]
```

- Đặc biệt, các thành phần trong list có thể khác datatype. Ví dụ:

```
danh_sach = [1, 2, "Hello", "3.45", (10, 11, 9), False]
```

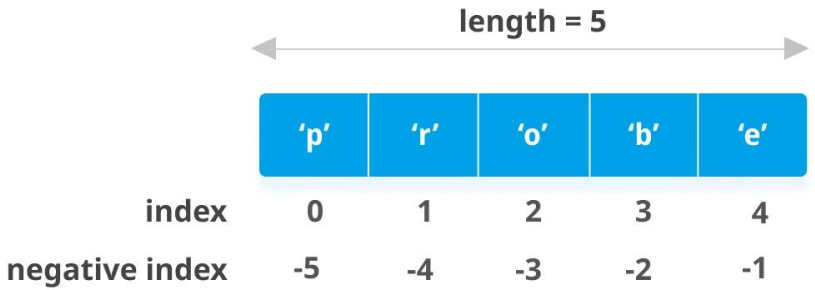
1.1 Giới thiệu

Các thành phần trong List là:

- Có thứ tự: mỗi thành phần có một thứ tự cố định
- Có thể thay đổi: có thể thay đổi giá trị của một thành phần, thêm hoặc bớt thành phần
- Có thể có giá trị giống nhau (duplicate values)

Các thành phần trong List có chỉ số (được indexed),

- Bắt đầu từ index [0]
- [-1] là thành phần cuối list,
  [-2] là thành phần áp chót…



1.1 Giới thiệu

Một “Array” (mảng) cũng là một biến (hoặc một cấu trúc dữ liệu) được dùng để lưu nhiều giá trị cùng lúc.

- Có thể hiểu một list cũng là một array
- Có 2 cách dùng array trong Python:
  - Sử dụng array module: yêu cầu các thành phần trong array phải cùng datatype
  - Sử dụng numpy: các thành phần có thể khác datatype

```
import array as arr
```

```
import numpy as np
```

Dùng array module

```
array_1 = arr.array("i", [3, 6, 9, 12])
print(array_1)
print(type(array_1))
```

```
array('i', [3, 6, 9, 12])
<class 'array.array'>
```

Dùng numpy

```
array_2 = np.array(["numbers", 3, 6, 9, 12])
print (array_2)
print(type(array_2))
```

```
['numbers' '3' '6' '9' '12']
<class 'numpy.ndarray'>
```

List	Array
Để sử dụng list không cần khai báo trước	Để sử dụng Array, phải khai báo trước
List thường không thuận tiện để lưu dữ liệu một cách chặt chẽ và với lượng lớn	Array có thể lưu dữ liệu chặt chẽ hơn (compactly), phù hợp để lưu lượng lớn dữ liệu
Thường ít được dùng để tính toán dữ liệu số	Array phù hợp hơn để thực hiện tính toán trên dữ liệu số
List được coi là chỉ có 1 chiều	Array có thể có nhiều chiều Khi tạo array nhiều chiều thì mỗi item phải có số lượng sub-items giống nhau

```
1 xyz = np.array([[1,2,3],["Hello", "Xin Chao", "Hi"]])
2 print(xyz)
```

```
[['1' '2' '3']
 ['Hello' 'Xin Chao' 'Hi']]
```

1.2 Các thao tác cơ bản

Tổng hợp các methods (phương thức) trên list:

- List index()
- List append()
- List extend()
- List insert()
- List remove()
- List count()
- List pop()
- List reverse()
- List sort()
- List copy()
- List clear()

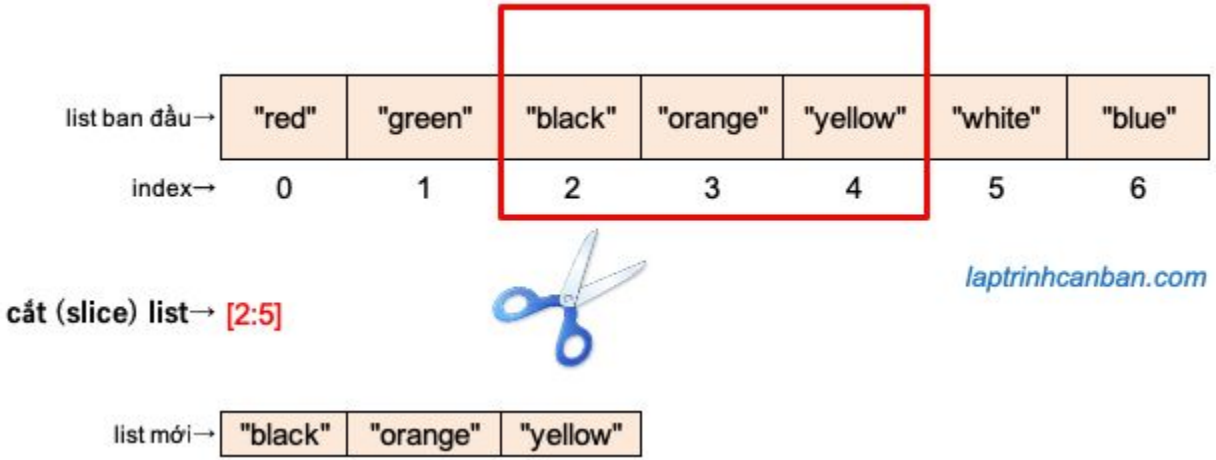
1.2 Các thao tác cơ bản

Các loại thao tác:

1. Khởi tạo & đếm số phần tử
2. Indexing và slicing
3. Đảo ngược list

1.2 Các thao tác cơ bản

- Indexing (thao tác trên chỉ số)
- Slicing (thao tác "cắt lát")
  cú pháp: a_list[start:stop:step]



Reversing (thao tác đảo chuỗi):

- Phương thức list.reverse()
- Dùng hàm reversed()
- Dùng slicing ngược

  a_list[::-1]

  a_list[-1::-1]

1.3 Thêm / Xóa / Sửa

Thêm vào List:

- Dùng phép cộng list
- Dùng hàm insert()
- Dùng hàm append()
- Dùng hàm extend()

Xóa khỏi List

```
# Xóa
ds_học_viên.remove("Mai")
print(ds_học_viên)

ds_học_viên.pop(3)
print(ds_học_viên)

del(ds_học_viên[4])
print(ds_học_viên)
```

```
['Nam', 'Lan', 'Long', 'Việt', 'Peter', 'Alex', 'Yeong-ho', 'Mai']
['Nam', 'Lan', 'Long', 'Peter', 'Alex', 'Yeong-ho', 'Mai']
['Nam', 'Lan', 'Long', 'Peter', 'Yeong-ho', 'Mai']
```

Sửa giá trị 1 item trong List

```
# Sửa
ds_học_viên[4] = "Yeong-hoon"
print(ds_học_viên)
```

```
['Nam', 'Lan', 'Long', 'Peter', 'Yeong-hoon', 'Mai']
```

1.4 Tìm kiếm

- Tìm Min/Max:
  - Hàm 'min'
  - Hàm 'max'

- Tìm kiếm một item nào đó:
  - Hàm List.**index()**
  - Toán tử 'in'

1.5 Sắp xếp

- Sắp xếp tăng
- Sắp xếp giảm
- Sắp xếp ngẫu nhiên

```
# Sắp xếp tăng
ds_học_viên = sorted(ds_học_viên)
print(ds_học_viên)

# Sắp xếp giảm
ds_chi_tieu = sorted([5, 6, 3, 7, 3, 8], reverse=True)
print(ds_chi_tieu)
```

```
['Alex', 'Lan', 'Long', 'Mai', 'Nam', 'Peter', 'Việt', 'Yeong-ho']
[8, 7, 6, 5, 3, 3]
```

1.5 Sắp xếp

- Sắp xếp ngẫu nhiên

  Sử dụng hàm
  - **random.shuffle**: sắp xếp list hiện tại
  - **random.sample(list, k)**: trả về một list mới với k phần tử được sắp ngẫu nhiên
- Ví dụ: chọn ra danh sách 100 khách hàng để gọi điện/tặng quà

1.6 Khái niệm List Comprehension

Cú pháp: `[tính-toán(item) for item in list]`

Ví dụ:

```
ds = ['Alex', 'Lan', 'Long', 'Mai', 'Mai', 'Nam',
      'Peter', 'Việt', 'Yeong-ho']

[len(ten) for ten in ds]
```

Có thể thêm điều kiện để lọc các item

```
[tính-toán(item) for item in list if điều-kiện]
```

1.7 Ghép 2 List với hàm `zip`

Hàm zip() thực hiện ghép 2 list theo cặp phần tử tương ứng vị trí với nhau

Ví dụ:

```
L1 = [9,3,5,7]
L2 = [4,2,8,6]
L  = zip(L1, L2)
```

Phần 2. Đọc/ghi dữ liệu (đơn giản)

2.1 Đọc/ghi file text

Plain text (.txt file):

- Plain text là định dạng file văn bản đơn giản
- Không sử dụng các format về màu sắc, trang trí, kiểu chữ đậm/nghiêng
- Có thể lưu dữ liệu phi cấu trúc
- Kích thước không lớn
- Trong quá trình phân tích dữ liệu:
  - Lưu dữ liệu thô
  - Lưu tạm dữ liệu trung gian
  - Lưu kết quả

2.1 Đọc/ghi file text

Các bước thực hiện:

1. Để đọc/ghi file, trước hết cần mở/tạo file theo chế độ (mode) tương ứng
2. Thực hiện thao tác đọc/ghi:
   - Các hàm Đọc: read(), readline(), readlines()
   - Các hàm Ghi: write(data), writelines()
3. Sau khi đọc/ghi xong, cần thực hiện đóng file bằng lệnh close()

Mode	Mô tả
'r'	Mở file chỉ để đọc
'w'	Mở file mới (ghi đè file cũ cùng tên)
'a'	Mở file đã có để ghi tiếp

2.1 Đọc/ghi file text

Câu lệnh mẫu “f = open(file_path, mode, encoding)”

```
f = open('danh_sach_hv.txt','r')
lines = f.readlines()
f.close()
print(lines)
```

```
danh_sach = ['Trần Văn A', 'Lê Văn B']
f = open('danh_sach_hv.txt','w')
for line in danh_sach:
    f.write(line)
    f.write('\n')
f.close()
```

2.1 Đọc/ghi file text

Câu lệnh mẫu "with open(file_path, mode, encoding) as …":

- Thường hay dùng trong Python
- Không phải gọi f.close()

```
lines = ['Trần Văn A', 'Lê Văn B']
with open('hoc_vien.txt', 'w') as f:
    for line in lines:
        f.write(line)
        f.write('\n')
```

2.1 Đọc/ghi file text

Câu lệnh mẫu "with open(file_path, mode, encoding) as …":

```
with open(file_path) as f:
    contents = f.readlines()
```

```
with open('hoc_vien.txt', encoding='utf8') as f:
    for line in f:
        print(line)
```

2.2 Đọc/ghi file csv

Định dạng CSV (comma separated values)

- Là một định dạng phổ biến trong việc lưu trữ, chia sẻ dữ liệu dạng spreadsheet (như trong Excel) và cơ sở dữ liệu.
- File .csv thật ra là dạng file text, đơn giản hơn so với file Excel
- Các bảng spread-sheet trong Excel đều có thể lưu lại dưới dạng file .csv

	A	B
1	production_budget_usd	worldwide_gross_usd
2	1000000	26
3	10000	401
4	400000	423
5	750000	450
6	10000	527
7	1800000	673
8	1000000	703
9	6600000	828
10	1000000	884
11	7000	900
12	2000000	926
13	1000000	1036
14	700000	1160
15	200000	1217
16	9000000	1242

File Excel

```
1 production_budget_usd,worldwide_gross_usd
2 1000000,26
3 10000,401
4 400000,423
5 750000,450
6 10000,527
7 1800000,673
8 1000000,703
9 6600000,828
10 1000000,884
11 7000,900
12 2000000,926
13 1000000,1036
14 700000,1160
15 200000,1217
16 9000000,1242
```

File CSV

2.2 Đọc/ghi file csv

Định dạng CSV (comma separated values)

Lưu ý: Việc sử dụng file csv vẫn chưa thật sự được chuẩn hóa, có nhiều cách vận dụng khác nhau. Cách tốt nhất là nên đơn giản hết mức có thể.

2.2 Đọc/ghi file csv

Mở file csv bằng Excel: có thể lỗi tiếng Việt (utf-8)

- Cách khắc phục: https://thuthuatphanmem.vn/sua-loi-file-csv-bi-loi-tieng-viet-khi-mo-bang-excel/

C	D	E	F	G
		DANH SACH		
		H? VÀ TÊN	NH.V?	
		Nguy?n vi?t an	TP	
		Tr?n qu?c bình	NV	
		Lê quang hà	BV	
		Bùi bích h?ng	G?	
		Nguy?n th? hùng	KT	
		Lý d?ng khiêm	TV	
		Võ thanh minh	NV	
		Lê hoàng nam	PG	
		Nguy?n kim oanh	KT	
		Nguy?n an s?n	PP	

2.2 Đọc/ghi file csv

Định dạng CSV (comma separated values)

- Cần sử dụng thư viện csv: `import csv`
- Mở file như mở .txt file
- Đọc/ghi thông thường: csv.reader(file)/csv.writer(file)
- Đọc/ghi kiểu dict: csv.DictReader(file), csv.DictWriter(file)

```
import csv
with open('hoc_vien.txt', 'r/w') as f:
    data_reader = csv.reader(f)
    data_writer = csv.writer(f)
    data_reader = csv.DictReader(f)
    data_writer = csv.DictWriter(f)
```

2.2 Đọc/ghi file csv

Nếu dữ liệu chứa dấu phẩy, khoảng trắng thì sao?

- `delimiter`: ký tự dùng để xác định dấu phân cách (thường là dấu phẩy)
- `quotechar`: khi dữ liệu chứa dấu phẩy, dùng single-quote, double-quote
- `escapechar`: ký tự sẽ bị bỏ qua
- `skipinitialspace`: bỏ qua space ngay sau dấu phân cách (True/False)

```
import csv

with open('hoc_vien.csv', 'r') as f:
    data_reader = csv.DictReader(f, delimiter='|',
                                 skipinitialspace=True,
                                 escapechar='-',
                                 quotechar='"')
    print(type(data_reader))
    for row in data_reader:
        print(row)
```

```
<class 'csv.DictReader'>
{'Tên': 'Nam', 'Tuổi': '22'}
{'Tên': 'MaiAnh', 'Tuổi': '25'}
{'Tên': 'Tùng', 'Tuổi': '32'}
{'Tên': 'Long', 'Tuổi': '28'}
```

2.3 Đọc/ghi file Excel

- Trong phân tích dữ liệu, việc cần thao tác với file excel thường sử dụng:
  - Đọc file Excel bằng công cụ `xlrd`: `import xlrd`
  - Ghi file Excel bằng công cụ `xlsxwriter`: `import xlsxwriter`
  - Sử dụng thư viện Pandas: thuận tiện hơn

⇒ Sẽ học cách đọc/ghi file Excel khi học đến phần Pandas

Phần 3. Một số (phân tích) thống kê đơn giản

3 Một số (phân tích) thống kê đơn giản

1. Measures of frequency: Number of Occurrences, Percentage
2. Measures of central tendency: Mean, Median, Mode
3. Measures of spread (dispersion/variabilty): Quartiles, Variance & Standard Deviation
4. Measures of position: Percentiles & Quantiles, Standard Scores
5. Measures of shape: Skewness/Kurtoris, Normal Distribution

(Các mục 3, 4, 5 sẽ thực hành sau, khi học đến thư viện numpy hay statistics và các thư viện nâng cao khác. Việc code bằng các hàm đơn giản sẽ mất nhiều thời gian)

3.1 Độ đo về tần suất

Độ đo về tần suất (Measures of frequency):
(Phân tích tần suất)

- Tần số (absolute frequency): số lần xuất hiện một giá trị trong tập dữ liệu
- Tần suất (relative frequency): tỷ lệ giữa tần số và tổng số lần xuất hiện của tất cả các giá trị trong dữ liệu (phần trăm số lần xuất hiện của từng giá trị)

```
# 3.1. Độ đo tần số:
# Hà Nội, Sài Gòn, Đà Nẵng, Đà Lạt, Quy Nhơn
ds_du_lich = ['HN', 'SG', 'ĐN', 'QN', 'HN', 'ĐL', 'ĐL', 'ĐN', 'QN', 'ĐL', 'QN', 'ĐN', 'ĐN', 'ĐL', 'QN', 'ĐL']
count_HN = 0
count_SG = 0
count_ĐN = 0
count_ĐL = 0
count_QN = 0

for tp in ds_du_lich:
    if tp == 'HN': count_HN += 1
    if tp == 'SG': count_SG += 1
    if tp == 'ĐN': count_ĐN += 1
    if tp == 'ĐL': count_ĐL += 1
    if tp == 'QN': count_QN += 1
```

3.2 Độ đo về khuynh hướng tập trung

Độ đo về khuynh hướng tập trung (Measures of central tendency):
(Phân tích giá trị trung tâm)

- Mean - The average value (giá trị trung bình)
- Median - The midpoint value (giá trị chính giữa, trung vị)
- Mode - The most common value (giá trị xuất hiện nhiều nhất)

THANK YOU!

Bài tập thực hành

1. Vui chơi với list. Viết mã để thực hiện:
   1.1. Chỉ lấy các phần tử ở vị trí chẵn trong chuỗi (0,2,4,6…,20)
   1.2. Chỉ lấy các phần tử ở vị trí lẻ trong chuỗi (1,3,5…,19)
2. Thực hiện lại các bài tập trong bài trước và lưu kết quả vào file text
3. Đọc file "**supermarket_sales_vn.csv**" và tính:
   3.1. **Tần số** & **Tần suất** khách hàng mua ở Tp HCM, Hà Nội, Đà Nẵng
   3.2. **Tần số** & **Tần suất** khách hàng Nam/Nữ
   3.3. Mean, median của Tổng đơn và Rating